Review of Agile SDLC for Big Data Analytics Systems in the Context of Small Organizations Using Scrum-XP

Gerardo Salazar-Salazar
Electronic System Department
Autonomous University of Aguascalientes, Mexico
gerardo.salazar@edu.uaa.mx

Manuel Mora
Information Systems Department
Autonomous University of Aguascalientes, Mexico
jose.mora@edu.uaa.mx

Hector Duran-Limon
Information Systems Department
University of Guadalajara, Mexico
hduran@cucea.udg.mx

Francisco Alvarez-Rodriguez
Computer Science Department, Autonomous University of Aguascalientes, Mexico
francisco.alvarez@edu.uaa.mx

Angel Munoz-Zavala
Statistics Department, Autonomous University of Aguascalientes, Mexico
eduardo.munoz@edu.uaa.mx

Abstract: *Software development using agile System Development Life Cycles (SDLC), such as Scrum and XP, has gained important acceptance for small businesses. Agile approaches eliminate barriers to required organizational, technical, and economic resources usually necessary when rigorous software development approaches, through heavyweight methodologies (e.g., Rational Unified Process (RUP)) or heavyweight international standards (e.g., ISO/IEC 12207) are used. However, despite their high popularity in small businesses, their utilization is scarce in the emergent domain of Big Data Analytics Systems (BDAS). Consequently, small businesses interested in deploying BDAS lack systematic academic guidance regarding agile SDLC for BDAS. This research, thus, addresses this research gap, and reports an updated comparative study of three of the main proposed SDLCs for BDAS (Cross-Industry Standard Process for Data Mining CRISP-DM), Two mains were Microsoft Team Data Science Process (TDSP), and Domino Data Science Lifecycle (DDSL)) in the current BDAS development literature, against a Scrum and Extreme Programming (Scrum-XP) SDLC. For this aim, a Pro Forma of a generic Scrum-XP SDLC is used to examine the conceptual structure, i.e., roles, phases-activities, roles, and work products-of these two SDLCs. Hence, this comparative study provides theoretical and practical insights on agile SDLC for BDAS adequate for small businesses and calls for further conceptual and empirical research to advance toward an agile SDLC for BDAS supported by academia and used in practice.*

Keywords: *Big data analytics systems, agile system development life cycle, Scrum-XP, CRISP-DM, TDSP and DDSL, small business.*

Received June 6, 2024; accepted October 9, 2024
https://doi.org/10.34028/iajit/21/6/12

1. Introduction

The agile Software Development Paradigm (SDP) emerged in the Software Engineering discipline about 20 years ago [32], as an alternative SDP to the dominant rigorous SDP [34] also known as plan-driven or heavyweight SDP -. Core literature on agile SDP [1, 23, 26, 32, 34] indicates that this paradigm was an overall response to address software development projects highly dynamic given changing user and system requirements, using new technological advances, and the business competitive pressures for shorting delivery timeframe from years to months. Additionally, there was also identified a strong disappointment with the current rigorous SDP because end-users and developers considered it a documentation-based bureaucracy that could be unnecessary for small software development projects [1, 32]. Consequently, formed an Agile Alliance consortium with several relevant practitioners [9] and declared the well-known Agile Manifesto that stands for one overall aim, four agile values, and twelve agile principles [9]. Table 1 reports these aims, values, and the twelve principles grouped in the categories of agile outcome, agile team, agile project, and agile design principles from [9, 56].

Nowadays, this agile SDP has permeated strongly in both small, medium, and large organizations [33, 34, 80] and co-exists with the rigorous SDP [7, 12, 48]. Several agile Software Development Life Cycles (SDLC) have been proposed [1, 34], but the most used and known at present days [22] are Scrum [74] and Extreme Programming (best known as XP) [8]. An SDLC refers to "the software processes used to specify and transform software requirements into a deliverable software product," [14]. An SDLC is usually represented as a software development process model [14] of phases-activities, roles, and work products proposed to increase the likelihood of delivering software on the expected

timeframe, on budget, and with expected functional quality, i.e., the named Iron Triangle [61]. A well-defined SDLC, either rigorous or agile, thus, "involves translating user needs into software requirements, transforming the software requirements into design, implementing the design in code, testing the code, and sometimes, installing and checking out the software for operational use." [14], and these activities can be performed sequentially, iteratively, or overlapped in a forward-backward way.

Table 1. Agile tenets (aim, values, and principles).

Type	Tenet
Aim	We are uncovering better ways of developing software by doing it and helping others to do it.
Values	• V1. Individuals and interactions over processes and tools. • V2. Working software over comprehensive documentation. • V3. Customer collaboration over contract negotiation. • V4. Responding to change over following a plan.
Principles	**Outcome Principles** • P1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software. • P7. Working software is the primary measure of progress.
	Project Principles • P2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage. • P3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
	Team Principles • P4. Business people and developers must work together daily throughout the project. • P5. Build projects around motivated individuals. Give them the environment and support they need and trust them to get the job done. • P6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation. • P8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely. • P12.At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.
	Design Principles • P9. Continuous attention to technical excellence and good design enhances agility. • P10. Simplicity-the art of maximizing the amount of work not done-is essential. • P11.The best architectures, requirements, and designs emerge from self-organizing teams.

Agile and lightweight SDPs, i.e., a balanced rigorous-agile type [12] such as the exemplary emergent ISO/IEC 29110 standard specifically designed for small organizations and very small entities [43] are usually considered equivalent. However, Agile and Lightweight SDPs are not theoretically equivalent [17, 62, 54]. Agile SDP can be considered a subset of the lightweight SDP [17, 62, 54]. Lightweight SDP can be defined as shortened phases-activities, roles, and work products from the original ones from rigorous SDP but are still considered useful for agile project domains. In contrast, the agile SDP, which also qualifies as lightweight, needs to be also flexible (i.e. to embrace changes), responsive (i.e., reactive to changes), rapid (i.e., applicable in relatively short periods), lean-seeking customer perceived (i.e., right-sized minimum viable product), simple (i.e., short-time training-learning periods), and with single fine-grained workflow guidance [17, 62, 54].

Agile SDP has been more frequently used by small and very small entities in the range of 5 to 10 people [2, 4, 22, 39, 60]. For instance, in [60], based on previous core literature on Software Process Improvement efforts for small and very small software development organizations, it was reported these types of organizations promote inherently agile-alike practices (constant face-to-face verbal communication rather than written documents, flexible, dynamic, and lightweight managerial practices, and flat organizational structures). In [2], it was found in a survey of 471 agile projects that the project size is negatively associated with software process success, and thus, the small-size projects usually developed by small and very small software development organizations use agile practices. In [4], from a conceptual review of the main agile SDPs, it is reported that whereas there are several ones, all of them can be applied for small and very small organizations with reduced development budgets, and scarce technological, organizational, and human resources. Finally, in [39], it was found that in 88 experience reports from software development start-up organizations, they inherently apply many agile practices such as "iterative development, empowered small team, and ongoing planning."

However, despite the extensive availability and utilization of the agile SDP-through a specific agile SDLC such as Scrum [74] or XP [8] or combined Scrum-XP [71, 76] for small and very small business software development organizations, for the emergent type of Big Data Analytics Systems (BDAS) their utilization has been reported as very scarce [30, 40, 42, 47, 49, 67, 68]. For instance, a Conceptual Review study [47] reported a research bias on BDAS algorithms, platforms, languages, and applications, but minimal on the Big Data Software Engineering (BDSE) area where SDLCs can be proposed. Later, in three studies using Systematic Literature Review or Systematic Mapping methods [40, 42, 68], it is reported that there is initial research on BDSE, but it is partial and focused on phase-activities of a generic SDLC rather than on a complete SDLC for BDAS. In [40], it was found that most available literature concentrated on proposing BDAS frameworks and architectures rather than full SDLCs. In turn [68], it was reported a set of critical success factors for BDAS projects, being one of them, the software development process model, and thus its implicit realized SDLC through a methodology or standard. In [42], it was reported that Architecture Design is the most published research on phases of SDLC for BDAS but again minimal on full SDLC. In [49], a Systematic Literature review found 19 SDLCs of type heavyweight, lightweight, and agile but reported

that only a few can be considered almost complete SDLCs, including project management, team management, and data management roles, activities, and products. Two mains were Microsoft Team Data Science Process (TDSP) [51] and Domino-Data Science Lifecycle (Domino DSL) [24], which have been classified as agile [66] and lightweight, respectively [52]. In [30], single case study research conducted in a large international company reviewed the agile Microsoft TDSP [51] and the classic Cross-Industry Standard Process for Data Mining (CRISP-DM) [16] SDLC and reported that both SDLCs need to be completed with missing activities and expected products from a BDSE perspective. Finally, in [67] a Systematic Literature Review conducted in the 2015-2021 period, 68 primary studies were collected, and it was reported that despite one of the main topics addressed was agile SDP, they only were focused on arguing the benefits of using agile SDP rather reporting tested new agile SDLC for BDAS. This study [67] also confirmed that CRISP-DM [16] is the most used SDLC for BDAS, but it must be ad-hoc adapted by practitioners because it was designed before the emergence of big data projects. Consequently, in [67], it is concluded that no SDLC can be claimed as the fact standard for developing BDAS, and more research on agile SDLC specific to BDAS is required.

The research on and the practical availability of well-tested SDLCs and in particular agile ones for small and very small organizations for BDAS is relevant because they are new high-valued software systems that have provided decision-making benefits mainly in the domains of Marketing, Healthcare, Finance and Manufacturing [3, 82], but they are also complex software products because they require complex computational processing, storing, and networking resources to apply advanced algorithms on large or very large datasets [41, 57, 64], and relevant organizational, economic and human technical resources usually only owned by large enterprises [20]. Consequently, frequent failed BDAS projects are still reported [21, 63]. As it is reported in [21]: "It is becoming increasingly clear that deployment getting analytical and Artificial Intelligence (AI) systems fully and successfully implemented within organizations is becoming one of the most critical disciplines at all phases of a business data science project." Similarly, in [63], from a business management perspective, a core survey collected answers from 3,000 respondents working in 29 types of industries located in 112 countries and identified that only 10% of them claimed financial benefits despite the large investments done in big data projects. Hence, practitioners and academics in the domain of BDAS demand well-tested-agile ones for small and very small organizations SDLC designed an ex-professor to BDAS.

In summary, these studies [30, 40, 42, 47, 49, 67, 68] using conceptual review, systematic literature review, systematic mapping, or single case study research methods report that there are already several SDLCs proposed for BDAS, but:

1. None well-accepted, and systematic SDLC specific for BDAS has gained a relevant international acceptance.
2. Large and medium-sized organizations have used old rigorous CRISP-DM SDLC, but it was proposed before the technical and organizational requirements currently demanded by BDAS, and thus it must be ad-hoc adapted introducing additional project management and technical risks.
3. Initial agile SDLC has been proposed, but its extensive utilization is still scarcely reported in real-life projects. Hence, the previous research consulted [30, 40, 42, 47, 49, 67, 68] on SDLC for BDAS has provided valuable insights, but it was also identified that studies on agile SDLC for BDAS specifically adequate for small and very small organizations is a relevant knowledge gap in the literature.

Consequently, this research addresses this problematic situation and reports a Conceptual Review of three of the main SDLCs for BDAS found in the scientific and gray literature CRISP-DM [16], Microsoft TDSP [51], and Domino DSL [24]), against the SDLC structure of agile roles, agile phases-activities, and agile work products of a generic agile Scrum-XP SDLC.

For this aim, a Pro Forma of the generic agile Scrum-XP SDLC is used to examine the conceptual structure i.e., roles, phases-activities, and work products of the three selected SDLCs. A Pro-Forma [5] is a pre-defined template reporting a set of organized concepts used as conceptual lenses to verify the extent of convergence to them from conceptual entities of interest of study (in this research, the three SDLC for BDAS). Pro Forms have been used for similar conceptual reviews [5, 6].

This article continues as follows: Section 2 reports a summary of the Research Approach. Section 3 reports the Theoretical Background regarding BDAS and the generic agile Scrum-XP SDLC. In section 4 are reported the summaries of the review of the three SDLC for BDAS, and an overall evaluation of them. Finally, in section 5, the conclusions and recommendations for further research are presented.

2. Research Approach

This research applies a Conceptual Review research method [70]. This study does not apply a Systematic Literature Review research method [37] because its research purpose is not to provide a statistical-descriptive accountability of findings on SDLC for BDAS, but to provide a thorough analysis for a better understanding of the structure of the main SDLCs for BDAS, according to the consulted literature [30, 40, 42, 47, 49, 67, 68]. Then, sub-section 2.1. reports the research objective and the research questions. Sub-

section 2.2. reports the Conceptual Review research method.

2.1. Research Objective and Questions

In this research, it is used a structured research objective template adapted from [83]. The adapted template is as follows: Analyze <objects of study> with the purpose of <purpose> concerning their <quality focus> from the perspective of <perspective> in the context of <context>.

Consequently, the research objective is formulated as follows: "Analyze < the main agile SDLC for BDAS> with the purpose of <describing, comparing, and evaluating them> regarding their <alignment of roles, phases-activities, and work products against a generic agile Scrum-XP SDLC > from the perspective of <the BDAS academic community> in the context of < SDLC for BDAS reported in the main scientific and gray literature on BDAS>." Three specific research questions were also stated as follows:

- **RQ 1:** what is the high-level structure (roles, phases-activities, and work products) of the main selected SDLCs for BDAS?
- **RQ 2:** what is the degree of alignment in roles, phases-activities, and work products of the SDLC for BDAS identified in RQ.1, concerning the generic agile Scrum-XP SDLC?
- **RQ 3:** can the three analyzed SDLCs for BDAS be considered agile in conformance with the generic agile Scrum-XP SDLC?

2.2. Research Steps and Materials

Table 2 summarizes the four research steps and materials of the Conceptual Review research method [70] used in this study:

1. Research formulation.
2. Research design.
3. Research analysis and synthesis.
4. Research reporting.

- *Step* 1. Research formulation was reported in section 2.1.
- *Step* 2. The research design was carried out in three sub-steps:

  a) Selection of potential sources for the objects of study.
  b) Selection of the objects of study from the potential sources.
  c) Establishing the concept for the analysis.

For the first step, the research team identified three recent studies of comprehensive literature reviews published in high-quality journals (high-impact JCR), which mention and compare different development cycles of systems for big data projects [28, 49, 67]. Each of these articles has a significant number of citations per year: [49] with 122 citations, [67] with 30 citations, and [28] with 164 citations. Martinez *et al*. [49] analyzed 19 development cycles; Saltz and Krasteva [67] located 27 primary studies where three methodologies were the main ones; and Giray [28] identified 17 studies on software engineering in the life cycle for big data.

Table 2. Research steps and materials.

Step	Purpose	Materials
1. Research formulation	To state the expected research objective that delimits the scope of the research, and the research questions that focus on the knowledge gaps of interest.	• Research objective statement. • Research questions.
2. Research design	To agree with the sources to collect the materials regarding the objects of study, and to define the conceptual tool that will be used to analyze the objects of study.	• Sources of materials. • Documents of the objects of study. • Conceptual tool for conducting the analysis.
3. Research analysis and synthesis	To conduct the analysis and synthesis of findings, using the conceptual tool, on the objects of study.	• Structured schemas of findings. • Summary of findings. • Conclusions on findings.
4. Research reporting	To produce valid and visible results for academic venues and outlets.	• Research report.

Table 3. Set of the top 5 to SDLC for BDAS.

SDLC for BDAS	Type of SDLC for BDAS	Publication domain	Publication name	Type of publication	Publication IF	Publication year	Citations	Is the SDLC reported in other SLR studies?
KDD	-	Analytics Data Science	Communications of the ACM	JCR journal	22.7	1996	3,541	Martinez, Saltz
CRISP-DM	Heavyweight	Analytics Data Science	SPSS Inc. Website	Gray Literature	-	2000	2,017	Martinez, Saltz
TDSP	Agile	Analytics Data Science	Microsoft Azure Website	Gray Literature	-	2016	22	Martinez, Saltz
DDSL	Lightweight	Analytics Data Science	Domino Data Lab Website	Gray literature	-	2017	6	Martinez,
BDPL	Heavyweight	Software Engineering	IEEE IT PROF	JCR journal	2.590	2018	15	Saltz, Giray

With these three studies of potential sources for big data development cycles, the research team agreed to select the five most important development cycles reported in these three comprehensive articles. Table 3 reports the five selected methodologies: KDD, CRISP, TDSP, DDSL, and Big Data Project Lifecycle (BDPL) with descriptive data. The research team agreed to carefully analyze the five selected development cycles,

where it was found that two of the development cycles did not meet the necessary characteristics to be included in the research. A detailed review indicated that KDD [27] does not meet the characteristics of an SDLC (phases, roles, and artifacts), but is instead considered a quick guide for developing BDAS projects. Additionally, the BDPL development cycle [45] was discarded for being a heavyweight SDLC, as it is based on one of the most robust standards in software engineering (ISO/IEC 15288).

Thus, for step 2.b), the first selected SDLC was CRISP-DM [16], which was reported in the three previous studies [37, 49, 67] as the most used SDLC for BDAS-type projects, despite being considered a rigorous methodology. The second selected SDLC was Microsoft's TDSP [51], which is one of the few agile SDLCs claimed for BDAS found in modern literature [30, 49, 67]. Finally, the Domino DSL SDLC [24] was selected, and despite providing a lightweight and competitive development approach [52], it was deemed necessary to determine its level of agility to avoid misinterpretations.

The research team manually identified two other SDLCs: IBM ASUM [19] and Data Driven Scrum (DDS) [69], but they were not included in this research as both are rarely referenced today. However, both could be considered for future research. It is worth noting that this research focuses on agile SDLCs, not rigorous ones. Therefore, only agile and lightweight SDLCs are considered to avoid misinterpretations, except for CRISP-DM, which was included for the reasons previously mentioned.

Finally, in step 2.c), the research team also agreed to develop the Pro-forma of the agile SDLC for Scrum-XP, including roles, phases-activities, and work products.

- *Step* 3. Research analysis and synthesis was conducted by the two first researchers, and later reviewed by the third and fourth researchers. Finally, all five researchers agreed on the final version of the findings.
- *Step* 4. Research reporting, this article was written.

3. Theoretical Background

To obtain the main SDLC for BDAS, the research design was executed through a selective manual search of the SDLCs for BDAS collected in the main literature [30, 40, 42, 47, 49, 68].

3.1. Big Data Analytics Systems

The term ‘big data’ emerged in 1997 from NASA researchers Cox and Ellsworth [18], who were the first to refer to ‘Big Data’ as: “Visualization poses an interesting challenge for computer systems of computer systems: the data sets are often quite large, straining the capacity of main memory, local disk, and even remote disk, local disk, and even remote disk. We call this the big data problem.” This has led to great importance in the field of BDAS, which have become increasingly important for academic and business communities in recent decades. However, only large business organizations are the regular clients and end users of data science analytics projects, focusing on cost-effective big data platforms [58, 78]. Nowadays, the current buzz surrounding the utilization of BDAS systems can be attributed to the promotional initiatives of certain leading technological companies that invested in building the analytical market niche. Some academics and professionals have regarded “big data” as data stemming from various channels, including sensors, satellites, social media sources, photographs, videos, and signals from cell phones and GPS [36]. All these massive data sources must be managed in a consolidated and integrated manner for organizations to derive data-driven value from their computational processing [82].

BDAS has been primarily characterized by the 5Vs attributes [10, 46, 55, 59, 65]: Volume, Velocity, Variety, Veracity, and Value. Volume refers to a large amount of data that needs to be recorded and requires significant storage capacity. Velocity refers to the frequency of data generation and/or data delivery. Variety refers to the fact that big data comes from a wide range of sources and formats that can be structured, semi-structured, or unstructured. Veracity represents the high quality of the data; this indicates that data verification is essential, as erroneous data will hinder decision-making or lead analysts astray. Value is created when data is analyzed and acted upon correctly to generate benefits for organizations (cost reductions, profitability increases, and business efficiency metrics, among others). Value can be classified into value discovery (through exploratory actions to discover potentially valuable business ideas), value creation (through the internal use of BDAS to increase the commercial value of the company), and value realization (through the delivery of products to end users).

However, the distinction between small data and big data is recent. Before 2008, data was rarely considered in terms of “small” or “big” [38]. All data was what is now sometimes called "small data", regardless of its volume. Due to factors such as cost, resources, and difficulties in generating, processing, analyzing, and storing data, limited volumes of high-quality data were produced through carefully designed studies using sampling frameworks designed to ensure representativeness [38]. In the case of Small and Medium-Sized Enterprises (SMEs), they are often hindered from reaping the benefits of using Analytics Data Science projects. Analytics Data Science approaches can also be applied to big data projects the size of small businesses [29, 35, 38, 75, 77]. In this research, we propose to distinguish between BDAS for large enterprises and BDAS for SME. Making this distinction is useful because the attribute of value

mentioned earlier in the 5Vs is of paramount importance for organizations, as it can generate value and competitive advantages with low-volume, low-velocity, and low-variety data, tailored to small business projects [29, 35, 38, 75, 77]. As a result, Table 4 illustrates the typical range of characteristics of the 5Vs among BDAS for large and small enterprise projects. It also includes a comparison between relevant complementary BDAS attributes: IT resources and IT personnel.

Table 4. Comparison between BDAS for large and small-medium business projects.

Attribute	BDAS for large business projects	BDAS for small-medium business projects
Data volume	Number of records from millions to billions or more. Size datasets from TBs to PBs or more. Datasets must be stored in a cluster of data servers.	Number of records from thousands to millions. Size datasets from MBs to TBs. Datasets can be stored in a single data server.
Data variety	The data sets contain structured, semi-structured and unstructured data (business records, xml/json texts, text documents, binary images, binary audios, binary videos, and binary streams).	Datasets contain structured data. Datasets sources are mainly internal business On-Line Transaction Processing (OLTP) and Data Marts systems. Datasets are recorded using Structured Query Language (SQL) and relational technologies.
Data velocity	Data arrives at very fast speeds; huge amount of data gets accumulated within a short period of time.	Controlled and steady flow of da-ta, accumulation of data is Slow.
Data veracity	From moderate to high data quality due to the main utilization of unstructured external business data sources. An Extract, Transform, Load (ELT) is used.	Very high data quality due to the main utilization of structured internal business data sources.
Data value	There is an implicit and potential utility value due to it not mandatory to count with the datasets for supporting business processes.	There is an explicit and current utility value due to the need to count with the datasets for supporting business processes.
IT resources	Moderate to large, distributed processing-storage server cluster.	Usually, a centralized single or a small processing-storage server cluster.
IT people	High-skilled on analytics, big data science, and big data IT services.	High-skilled on analytics.
BDAS exemplary case	Uninterruptible Power Supply (UPS) now tracks data on 16.3 million packages per day for 8.8 million customers, with an average of 39.5 million tracking requests from customers per day. The company stores over 16 petabytes of data [84].	A predictive analytics approach for myocardial infarction was developed using statistical techniques. Several Machine Learning (ML) models were also applied. The dataset included 47,786 records with 21 input features and 1 output one, and it was about 3 MB [13].

Projects for developing BDAS, both small and large, are technically challenged to be successful [21, 63]. Various international reports indicate that a significant percentage of BDAS projects failed to be completed within budget, schedule, or planned functionality [21, 63]. Agile development methodologies have been proposed for BDAS to address the issue of failed projects [44, 79]. Therefore, in this research, we will compare Scrum-XP, which is one of the most widely used agile methodologies [22], against three of the best-known methodologies for BDAS projects: CRISP-DM [16], Microsoft TDSP [51] and Domino DSL [24].

3.2. A Pro-Forma of the Theoretical Generic SCRUM-XP SDLC for BDAS

With the introduction of the four values and twelve principles of the Agile Manifesto [32], the term SDP gained momentum, strongly permeating and coexisting with the traditional rigor-oriented development paradigm [11]. The fundamentals and principles of the manifesto enabled the development of methods with a real-world focus, where responsiveness to change became a key factor for success [15]. Because of the introduction of the Agile Manifesto for software development over two decades ago [32], agile methodologies were created to improve software development and counter traditional processes, where software projects were characterized by low flexibility, long delivery times, excessive documentation, bureaucratic processes, and usually costs overruns.

Two of these methodologies are Scrum and XP, which, in the last decade, have emerged as two of the most widely used Agile SDLCs for software development [22]. Sutherland [74] define the Scrum as an "iterative and incremental framework for projects and product or application development." On the other hand, XP is defined as an unconventional software process that "rather than planning, analyzing, and designing for the far-flung future, XP exploits the reduction in the cost of changing software to do all of these activities a little at a time" [8]. This has led some authors [71, 76] to suggest combining both methodologies to create a hybrid Scrum-XP methodology, to complement the agile management of activities handled by Scrum and the engineering processes contemplated by XP.

Recently, the literature has summarized and updated [15, 54] the core literature on the agile SDP [1, 17, 62] and the Scrum [53, 74] and XP [8, 72] development methods to a generic integrated Scrum-XP SDLC, to a set of 6 agile values, and a rigorous-agile SDLC framework of 7 attributes. The six agile values refer to:

1. Individuals and interactions over processes and tools
2. Working software over comprehensive documentation.
3. Collaboration of the entire team over contracts.
4. Responding to change over following plans.
5. The process is maintained and perceived as agile.
6. The process is cost-effective.

Table 5. Pro forma of the agile Scrum-XP SDLC for BDAS (from core agile software literature).

SDLC element	SDLC element description
Roles (3)	• User roles: R.1: Scrum-XP product owner. • Management roles: R.2: Scrum-XP master. • Technical roles: R.3: Scrum-XP development team.
Phases-activities (6, 13)	**Pre-Game Phases:** • Phase 1. Product exploration: to obtain the user requirements through the initial (no prioritized) and final (already prioritized) full product backlog (user stories) work product. If required, to explore empirically a Spike. Activities: 1. Product vision declaration. 2. Product backlog (user stories) elaboration and prioritization. 3. Spikes exploration (if required). • Phase 2. Product release planning: to elaborate an agreed product backlog (user stories) development plan. Activities: 4. Product backlog (user stories) development planning.
	Game Phases: • Phase 3. Sprint-iteration planning: to elaborate an agreed Sprint-Iteration backlog (user stories) development plan. Activities: 5. Sprint-Iteration backlog (user stories) development planning. • Phase 4. Sprint-iteration development**:** to sketch a simple architectural design supported by the current Sprint-Iteration backlog (user stories), build the Sprint-Iteration backlog (user stories). Activities: 6. Simple architectural design. 7. Daily Scrum-XP meeting. 8. User acceptance tests elaboration. 9. Technical tests elaboration. 10. Increment building, testing and integration. • Phase 5. Sprint-iteration review and retrospective: to conduct the Sprint-Iteration review and retrospective. Activities: 11. Sprint-Iteration review. 12. Sprint-Iteration retrospective.
	Post-Game Phase: • Phase 6. Product release: to deliver the final. Work products: 14. Software product release. Activities: 13. Product release delivery.
Work products (15)	**Pre-Game Phases:** • Phase 1. Product exploration. Work products: 1. Product vision statement. 2. Product backlog (user stories). 3. Spike records (if used). • Phase 2. Product release planning. Work products: 4. Product backlog (user stories) development plan.
	Game Phases: • Phase 3. Sprint-iteration planning. Work products: 5. Sprint-Iteration backlog (user stories) development plan. • Phase 4. Sprint-iteration development. Work products: 6. Simple architectural design. 7. Daily Scrum-XP 3-question record. 8. Kanban board. 9. Burndown chart. 10. User acceptance tests. 11. Technical functional tests. 12. Sprint-Iteration software increment. 13. Sprint-Iteration software build. • Phase 5. Sprint-iteration review and retrospective. 14. Sprint-Iteration review record.
	Post-Game phase: • Phase 6. Product release. Work products: 15. Software product release.

The seven expected attributes for agile SDLC are:

1. Flexible (i.e., can be reconfigured when necessary).
2. Responsive (i.e., promote detection and reaction to changes).
3. Fast (i.e., applicable in relatively short periods).
4. Lean (i.e., generate a minimum viable software product).
5. Simple (i.e., have low cognitive load and training effort to be learned and used).
6. Lightweight (i.e., documented in few pages).
7. Optional documentation (i.e., Development Team decides detail level of technical documentation elaborated).

For this set of 7 agile attributes, the seven rigorous counterparts are:

1. Rigid.
2. Bureaucratic.
3. Slow.
4. Sophisticated.
5. Hard.
6. Heavy.
7. Mandatory documentation.

Figure 1 depicts this theoretical generic Scrum-XP SDLC adapted from the literature on agile software development [1, 8, 25, 62, 72, 74]. Table 5 reports this SDLC in a Pro Forma design of roles, phases-activities (grouped into three classic phase categories named Pre-Game, Game, and Post-Game [73]), and work products. This Scrum-XP SDLC for BDAS has three roles, 6 phases including 13 activities, and produces 15 work products.

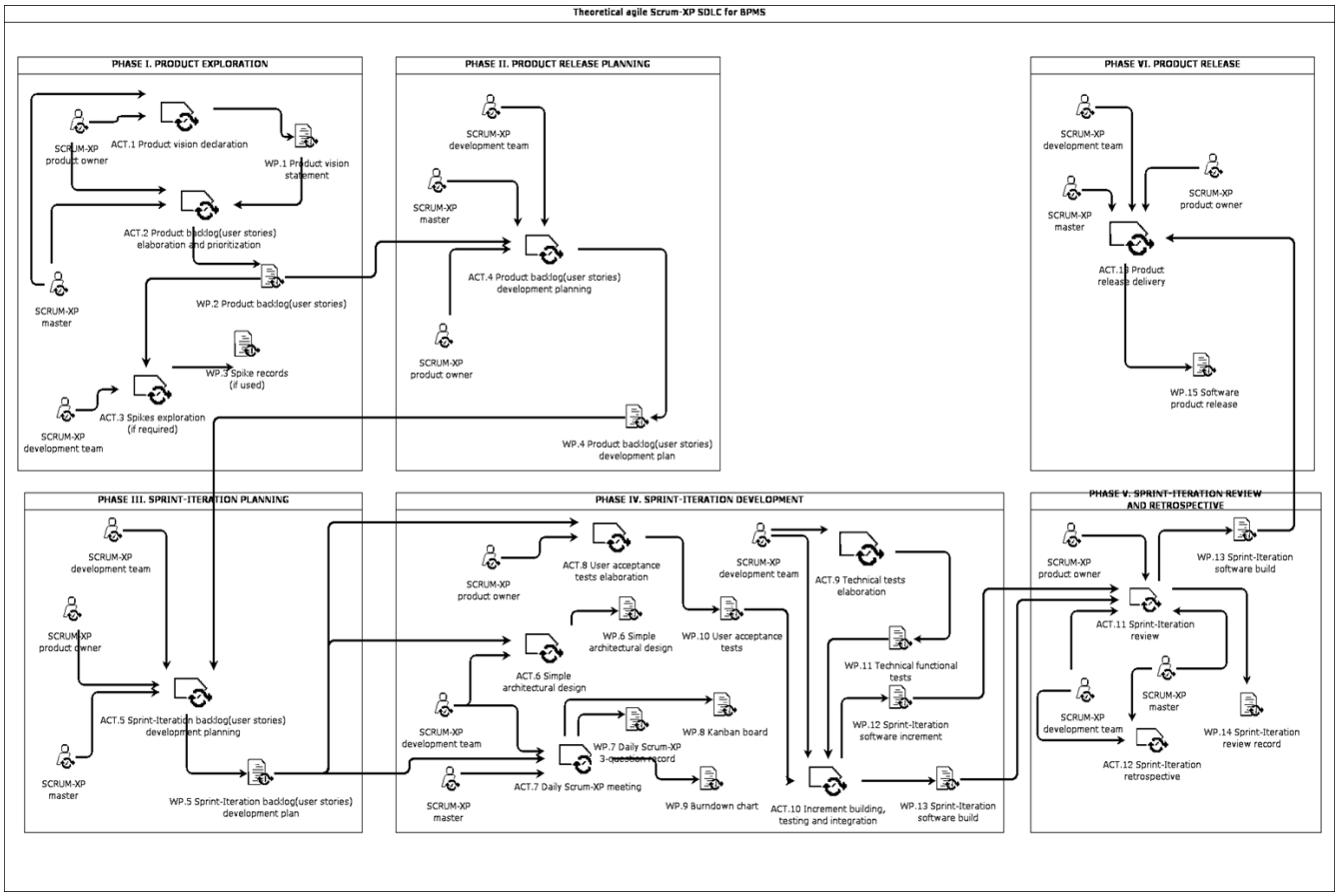

Figure 1. The theoretical generic agile Scrum-XP SDLC for BDAS (from core agile software literature [1, 8, 25, 62, 72, 74]).

4. Analysis and Discussion of Comparative Results on the three SDLCS for BDAS

In this section, we report the analysis of the two BDAS SDLCs found in the literature, detailing their roles, phases-activities, and work product structure, to compare against the previously established generic Scrum-XP SDLC Pro-Forma. These methodologies are:

- CRISP-DM [16].
- TDSP [51].
- DDSL [24].

4.1. Analysis of the CRISP-DM SDLC

In 1999, the first edition of the CRISP-DM standard procedure, an acronym for CRISP-DM, was introduced [16, 27, 50]. CRISP-DM was created to categorize and guide the most common steps in data mining projects. It quickly became consolidated as “the de facto standard for developing data mining and knowledge discovery projects” [31] and remains to this day the most widely used data analytics methodology according to various opinion surveys [81]. The CRISP-DM methodology provides a structured approach for planning and

developing a BDAS-type project and has served as the basis for the creation of other SDLCs [50].

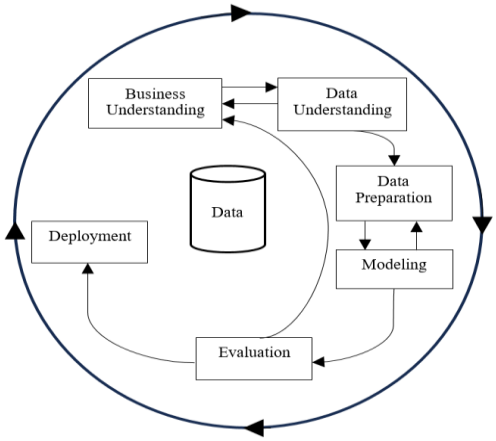

Figure 2. The CRISP-DM SDLC for BDAS. Take to [16].

The life cycle proposed by the CRISP-DM SDLC consists of 6 phases, 24 activities, and 42 work products [16], initially sequential but typically performed retrospectively. The authors do not mention the roles required to manage such projects, but for this research, standard roles that would be used in any BDAS project will be considered (customer, project manager, and data science development team). Below are described the 6 phases with their respective activities and work products according to the CRISP-DM methodology, and Figure 2 illustrates this SDLC for BDAS.

- *Phase* 1. Business understanding: in the initial stage, we focus on understanding the project's objectives and requirements. Once obtained, this information is transformed into a definition of the data mining problem and a preliminary plan to achieve those objectives. For this first phase, there are four activities:

  1. Determine business objectives: the main objective of this activity is to thoroughly understand, from a business perspective, the client's requirements.
  2. Assess situation: this activity involves conducting a detailed investigation of all resources, limitations, assumptions, and risks, among other factors considered to determine the project's objective.
  3. Determine data science goals: this activity establishes the project's objectives in technical terms.
  4. Produce project plan: this activity aims to produce a plan to achieve the data science goals and thus achieve the business objectives. This plan should include the planned steps to be carried out throughout the rest of the project.

- *Phase* 2. Data understanding: this stage begins with the collection of information and proceeds with actions to delve into the data, identify data quality issues, discover early insights from the data, or detect interesting subsets of data. This phase consists of 4 activities:

  1. Collect initial data: this process involves acquiring datasets, the location where they are stored, and the methods used to acquire them.
  2. Describe data: its objective is to examine the "raw" or "superficial" properties of the acquired data and report the results.
  3. Explore data: data exploration helps address data extraction issues, considering assumptions and their impact on the rest of the project. This process can be approached through queries, visualization and reporting, and statistical analysis, among others.
  4. Verify data quality: in this phase, questions such as "Are the data complete (covering all necessary cases)?" "Are they correct, or do they contain errors, and if so, how often?" "Are there missing values in the data? If so, how are they represented, where do they occur, and how often?" are addressed.

- *Phase* 3. Data preparation: this phase encompasses all actions aimed at creating the definitive dataset from the raw dataset. Tasks include selecting tables, records, and attributes, as well as transforming and cleaning the data for modeling tools. This phase includes 5 activities:

  1. Select data: in this phase, the data to be used for analysis will be decided. It includes selection criteria such as relevance to the objectives, quality, and technical limitations, as well as limits on volume or data types.
  2. Clean data: the main objective of this activity is to improve data quality, representativeness, and impartiality. This may involve selecting clean subsets of data, inserting appropriate default values, or more ambitious techniques such as estimating missing data through modeling.
  3. Construct data: data construction is the process of developing new records or producing derived attributes.
  4. Integrate data: this stage provides methods by which information from various tables or records is combined to create new records or value scores.
  5. Format data: it focuses on syntactic modifications made to the data without changing its meaning.

- *Phase* 4. Modeling: during this phase, various modeling techniques are chosen and applied. Typically, there are multiple methods to address the same type of data science problem. Phase 4 consists of 4 activities:

  1. Select modeling techniques: specific modeling techniques are selected to be applied to the datasets. Different modeling techniques can be applied to the same dataset.

2. Generate test design: tests are generated to determine the robustness, quality, and validity of the model before building it.
3. Build model: select models are implemented and parameterized on the prepared dataset.
4. Assess model: model evaluation focuses on interpreting the model based on quality metrics, project success criteria, desired test design, and data science results in the business context.

- *Phase* 5. Evaluation: before proceeding with the final implementation of the previously created model, it is crucial to conduct comprehensive evaluations of the developed model and carefully review the steps followed for its construction. This ensures that the model adequately meets the business objectives and requirements established in Phase 1. For Phase 5, we have 3 activities:

  1. Evaluate results: in this stage, the degree to which the model meets the business objectives is assessed, and attempts are made to determine if there are any business reasons why this model may be deficient.
  2. Review process: this activity focuses on quality assurance by analyzing all steps to ensure that the project covers all business issues.
  3. Determine next steps: the next steps are determined based on the evaluation results and process review. The data science team decides whether to implement the models, conduct additional iterations to improve the results, or conclude the project without actual implementation.

- *Phase* 6. Deployment: this stage varies depending on the requirements of the data science project and can range from generating reports to implementing a repeatable data mining process. Generally, the completion of the model does not mark the end of the project. In Phase 6, we have 4 activities:

  1. Plan deployment: specific actions and resources are established to implement the models or their results.
  2. Plan monitoring and maintenance: strategies are defined and agreed upon to keep the implemented models or their results alive.
  3. Produce final report: the final documentation for the Client is generated, and this activity concludes the Data Science project.
  4. Review project: the team analyzes the positive and negative events that occurred during the data science project to learn from them and avoid them in the future.

4.2. Analysis of the TDSP SDLC

TDSP is an iterative data science methodology based on the CRISP-DM methodology [51], which is self-claimed as agile. It is an SDLC for BDAS projects that pursues efficient BDAS applications [51]. TDSP provides guidelines and frameworks from its publisher company to facilitate the proper implementation of BDAS projects [51]. This SDLC proposes different roles, activities, and work products for the development of BDAS projects, which are very clear and assist in the creation, execution, and development of projects [51]. The roles mentioned in TDSP are well-defined, which helps improve collaboration and coordination within the team. This SDLC manages four roles: Customer, Project Manager (for the overall managerial coordination of the BDAS development project), Project Lead (for the technical coordination of the BDAS development project), and project individual contributors (solution architect, data engineer, data scientist, and application developers). The TDSP lifecycle for structuring the development of its projects consists of consists of 5 iterative phases, 14 activities, and 12 work products. Next, these phases-activities are described, and Figure 3 illustrates this SDLC for BDAS.

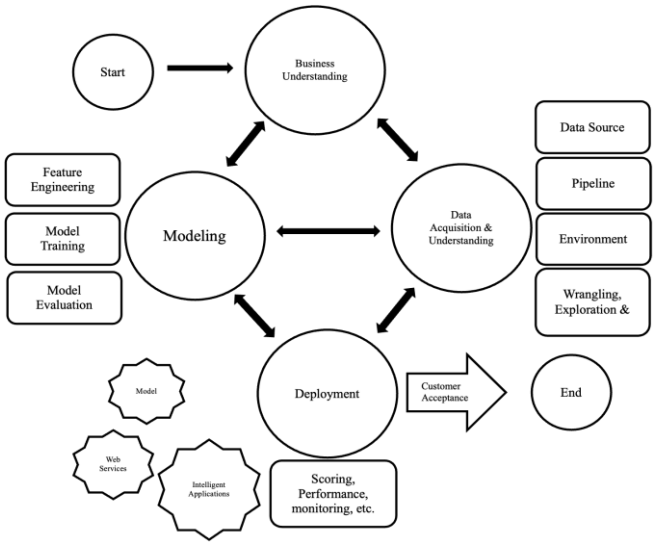

Figure 3. TDSP SDLC for BDAS. Take to [51].

- *Phase* 1. Business understanding: the objective of this phase is to identify the main variables that will serve as model objectives, and project success metrics, and to identify data sources. For this purpose, it includes 2 activities:

  1. Define objectives: the main objective is to identify the project's goals by interacting with the client and formulating core questions that data science can address. Additionally, defining the project team to carry out the project by specifying roles and responsibilities.
  2. Identify data sources: the required datasets for the BDAS that can help answer the Client's queries are defined. This phase also presents 3 work products:

     a) Charter document.
     b) Data source.

c) Data dictionaries.

Additionally, TDSP documentation reports a TDSP workflow for project execution with three activities: 3 plan sprint, 1. Review code built from several branches, and 5 merge-delete branches. These activities can be considered Phase 0. Agile project management.

- *Phase* 2. Data acquisition and understanding: in this phase, a clean and high-quality dataset is generated, and the data architecture solution is developed. It consists of 3 activities:

  1. Ingest the data: data is moved from source locations to destination locations where analysis operations are performed.
  2. Explore the data: datasets are explored and processed to remove noise, discrepancies, or missing data. This is done to create a high-quality dataset that will be used for project development.
  3. Set up a data pipeline: the data ingestion architecture is specified based on business needs and constraints (batch mode, streaming, real-time, or hybrid). Additionally, it includes 3 work products:

     a) Data quality report.
     b) Solution architecture.
     c) Checkpoint decision.

- *Phase* 3. Modeling: the data for the learning model is determined, and a ML model is created. This phase consists of 3 activities:

  1. Feature engineering: TDSP provides a methodological guide for selecting the most appropriate model (referred to as the ML algorithm reference sheet).
  2. Model training: in this part, ML models are trained and calibrated.
  3. Model evaluation: this activity determines whether the trained and calibrated statistical/ML model produces results with a level of validity suitable for use in production. For this phase, the authors do not report any work product.

- *Phase* 4. Deployment: in this phase, the models with data pipelines are implemented in a production environment. To achieve this, the phase consists of 1 activity:

  1. Operationalize the model: the main objective of this activity is the implementation of the model and the pipeline in a production or similar environment for application consumption. This activity includes 3 work products:

     a) Status dashboard that displays the system health and key metrics.
     b) A final modeling report with deployment details.
     c) A final solution architecture document.

- *Phase* 5. Customer acceptance: this phase aims to ensure the model and its implementation meet all customer requirements. This phase involves 2 activities:

  1. System validation: confirming that the implemented model and pipeline meet the customer's needs.
  2. Project hand-off: handing over the project to the entity that will execute the system in production. It includes 1 work product:

     a) Exit report of the project for the customer.

4.3. Analysis of the DDSL SDLC

The Domino DSL [24] is a recent full SDLC for BDAS classified previously as lightweight [52]. DDSL SDLC relies on three principles proposed for current BDAS contexts:

1. To be iterative.
2. To foster collaboration between the customer. Project manager, and development team.
3. To visualize and attend to future relevant organizational impacts of the developed BDAS. Consequently, DDSL SDLC [24] is structured into 3 roles, 6 and 35 phases, and activities, respectively, and 9 work products.

The phases are:

1. Ideation.
2. Data acquisition and exploration.
3. Research and development.
4. Validation.
5. Delivery.
6. Monitoring.

The roles are:

1. Business stakeholders (customers, users).
2. Data scientists.
3. IT team (data product manager, data storyteller, and data infrastructure engineers).

- *Phase* 1. Ideation: this phase focuses on establishing the business objectives for the planned BDAS, but a specific business problem must be previously selected, the economic and technical feasibility of the BDAS project is assessed, the BDAS requirements are documented, and the decision to advance to the next stage or abandon the project is made. There are 4 activities as follows.

  1. Project scoping: the business objectives of the BDAS are set up, and economical-technical feasibility is assessed, classifying the project as “sweet spot”, “transformational”, “quick wins,” or “don’t just don’t” types.
  2. Proceed decision: the go-no-go decision is agreed to continue or abandon the BDAS development project because it is not feasible to be developed.

3. Select artifacts: in the case of BDAS project continuation, an overall BDAS architectural solution. This phase 1. thus elaborates 2 work products:

   a) Requirements documentation.
   b) BDAS architectural solution.

- *Phase* 2. Data acquisition and exploration: this phase refers to the identification of the required available and non-available datasets, its financial-technical authorization for getting them, and its iterative exploration, pre-processing, and understanding for the next phase. There are 6 activities in phase 2.

  1. Identify datasets: access to internal datasets must be authorized by the IT department, and/or external datasets must be authorized to be bought.
  2. Ingest data: internal and/or external datasets are transferred from the original locations to the target BDAS location.
  3. Explore data: iteratively and interactively, the datasets are explored to determine the final ones to be used.
  4. Prepare data: pre-processing and processing operations on the final datasets to be used are applied. There is one work product in this phase:

     a) Data dictionary.

- *Phase* 3. Research and development: this phase refers to the selection, building, and calibration of the statistical/ML model, which also includes the selection of the data science and analytics platforms and tools to be used. There are 5 activities in this phase 3.

  1. Generate hypothesis and model: the set of specific inquiries is formulated, and the statistical/ ML model is selected activity.
  2. Validate right platforms and tools: where the computational development resources are already available or are requested by the IT department.
  3. Experiment and assess result: the statistical/ ML model is performed, and calibrated, and the results are assessed to determine whether they are sufficiently insightful to advance to the next phase or more datasets and experimentation-calibration is required. Phase 3 produces one work product:

     a) Calibrated statistical/ ML model.

- *Phase* 4. Validation: this phase refers to the business and technical validation of the Calibrated Statistical/ ML model to authorize its delivery to production or return to conduct required previous stages or to stop the BDAS development project because this does not reach the business reliability expectations. There are 3 activities in this phase 4.

  1. Business validation: business stakeholders determine whether the results of the BDAS model are useful and reliable from the business perspective.
  2. Technical validation: the IT team determines whether the BDAS model is ready for its deployment to production. There is one work product elaborated in this phase 4:

     a) BDAS business and technical validation documentation.

- *Phase* 5. Delivery: this phase focuses on becoming the statistical/ ML model in a "product" usable by customers and users. There are 4 activities in this phase 5.

  1. Plan delivery: a detailed plan for deploying the BDAS is conceived (among final modes such as an ad-hoc report, scheduled report, application launcher, web application, batch API, or real-time API).
  2. Deploy: it is applied to the selected deployment mode.
  3. Alpha/beta test: technical internal (Alpha) and pilot user (Beta) tests are applied.
  4. User acceptance test: an official group of users verifies the acceptance of the BDAS. In this activity, it is not expected that Users reject the BDAS, i.e., if the BDAS development project reached this activity, it is because it is a satisfactory product. There are two work products elaborated in this phase:

     a) Monitoring and training plan.
     b) Tests documentations (Alpha, Beta, and user acceptance types).

- *Phase* 6. Monitoring: this phase refers to the periodic supervision and evaluation of the usage, technical performance, and created value of the BDAS. This phase 6 has two activities.

  1. Supervise and Evaluate Usage and Performance: usual IT service managerial metrics can be applied to keep the BDAS usable.
  2. Evaluate Value: Business Stakeholders determine the overall and specific contributions of the BDAS to the business value. Additionally, Business Stakeholders can propose improvements to the BDAS for the next version of the installed and used product. Then, this last phase 6 generates two work products:

     a) Periodical usage and performance evaluation report.
     b) Overall value evaluation report.

The life cycle of DDSL consists of 6 phases, 18 activities, and 9 work products. Below, Figure 4 shows the life cycle of the DDSL SDLC.

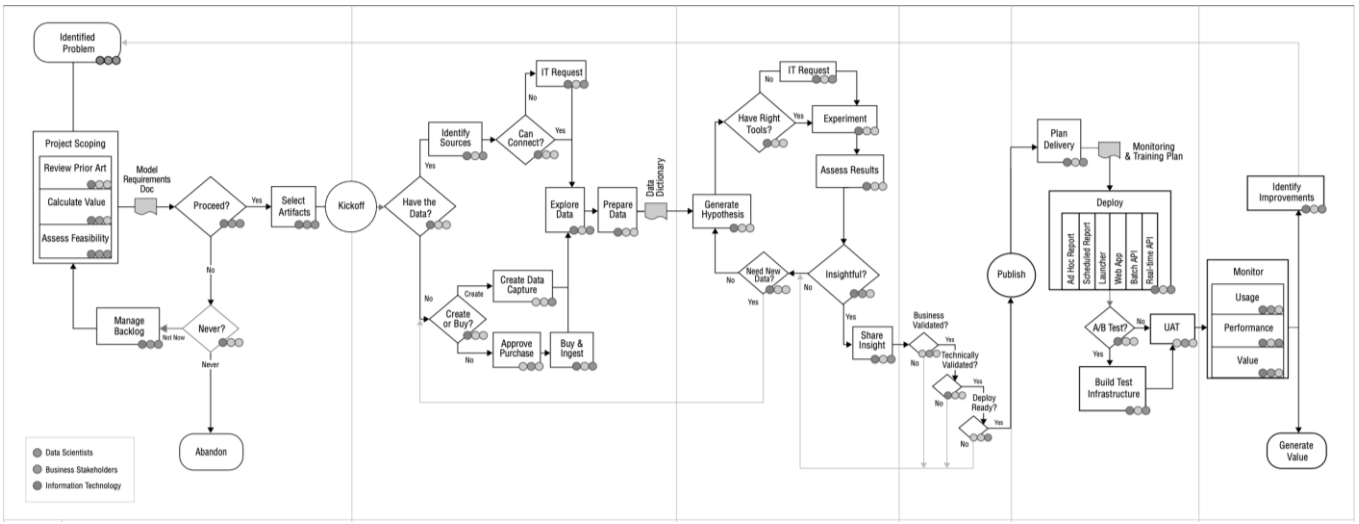

Figure 4. The DDSL SDLC for BDAS. Take to [24].

4.4. Integrated Analysis and Discussion on the Two SDLCs for BDAS Against the Agile Generic SCRUM-XP SDLC Pro Forma

For this research, a detailed review of three rigorous SDLCs was conducted, namely CRISP-DM [39], which is the most widely used methodology for developing BDAS projects [81], the TDSP methodology proposed by Microsoft [51], which is claimed as an agile methodology for BDAS project development, and a recent lightweight Domino DSL SDLC [24]. The structure and content of roles, phases-activities, and work products proposed in CRISP-DM, TDSP, and DDSL were analyzed against the expected roles, phases-activities, and work products of the theoretical generic Scrum-XP SDLC Pro Forma reported in sub-section 3.2.

Table 6. Evaluation of CRISP-DM, TDSP, and DDSL against the theoretical agile generic Scrum-XP SDLC pro-forma.

SDLC element	Theoretical agile generic Scrum-XP SDLC pro-forma	CRISP-DM	TDSP	DDSL
Roles (3)	User roles: 1. Management roles: 1. Technical roles: 1.	Low	Moderate	Moderate
	Overall evaluation of roles	Low level of roles	Moderate level of roles	Moderate level of roles
Phases-activities (6, 13)	Pre-game phases: • Phase 1. Product exploration: (with 3 Activities). • Phase 2. Product release planning: (with 1 activity).	Low	Low	Moderate
	Game phases: • Phase 3. Sprint-iteration planning: (with 1 activity). • Phase 4. Sprint-iteration development: (with 5 activities). • Phase 5. Sprint-iteration review and retrospective: (with 2 activities).	Moderate	High	Moderate
	Post-game phase: • Phase 6. Product release: (with 1 activity).	Low	High	Moderate
	Overall Evaluation of Phases-Activities	Low level of phases-activities	High level of phases-activities	Moderate level of phases-activities
Work products (15)	Pre-game phases: • Phase 1. Product exploration: (3 work products). • Phase 2. Product release planning: (1 work products).	Low	Moderate	Moderate
	Game phases: • Phase 3. Sprint-iteration planning: (1 work product). • Phase 4. Sprint-iteration development: (8 work products). • Phase 5. Sprint-iteration review and retrospective: (1 work product).	Moderate	Moderate	Low
	Post-game phase: • Phase6. Product release: (1 work product).	Low	Moderate	Moderate
	Overall evaluation of work products	Low level of work products	Moderate level of work products	Moderate level of work products
	Overall evaluation of SDLC	Overall low level of the SDLC	Overall moderate level of the SDLC	Overall moderate level of the SDLC

Based on these detailed analyses Table 8, a graphical summary of the review is presented in Table 6. The following qualitative scale of alignment and adherence of the analyzed SDLC and the theoretical generic Scrum-XP SDLC pro-forma was used:

- Low level (1 point) this corresponds to a cell shaded in light gray when the analyzed SDLC contains relevant omissions regarding the expected content on roles, categories of phases-activities, or categories of

work products (packages) of the theoretical SDLC for BDAS.
- Moderate level (3 points) this corresponds to a gray cell when the analyzed SDLC contains slight omissions regarding the expected content on roles, categories of phases-activities, or categories of work products (packages) of the theoretical SDLC for BDAS.
- High level (5 points) this corresponds to a dark gray cell when the analyzed SDLC contains relevant similarities regarding the expected content on roles, categories of phases-activities, or categories of work products (packages) of the theoretical SDLC for BDAS.

This review was carried out in several iterations and is based on the complete content reported in the sources. Content omissions, differences in interpretation, and typographical errors in the nomenclature of roles, phases-activities, and work products were identified, corrected, and agreed upon in a single evaluation by the research team.

Based on the integrated results reported in Table 5, the research team also evaluated the three SDLCs for BDAS concerning the theoretical rigorous-agile SDLC framework of 7 attributes (see Table 6) mentioned in section 3.2. of this research. Table 6 also includes the evaluation of the agile generic Scrum-XP SDLC (in the +3 zone). From these analyses (Tables 5 and 6), consequently, we can summarize the following findings on strengths and weaknesses for the three analyzed SDLCs for BDAS as follows:

Strengths

- The three SDLCs share the claim regarding the need for having an ex-professor SDLC for BDAS projects instead of developing projects without any guidance or a general-purpose SDLC. Additionally, the three SDLCs include activities specific to Data Acquisition, Exploration, and Understanding.
- The three SDLCs have been used in real-world BDAS projects.
- The CRISP-DM, despite reaching a low conformance level regarding the generic Scrum-XP SDLC and being assessed in the rigorous zone of SDLC (-2.0 of score), has been reported still as the most used SDLC for BDAS (but with adequations).
- The SDLC TDSP, as it was expected, reached the best conformance level (high) regarding the generic Scrum-XP SDLC, and it was assessed in the agile zone of SDLC (+2.0 of score).
- The most recent SDLC analyzed, DDSL, reached a moderate conformance level regarding the generic Scrum-XP SDLC, and was assessed in the lightweight zone of SDLC (+1.0 of score).

Weaknesses

- The three SDLCs are published by private organizations, and thus, their free-access public documentation is limited. Additionally, there are scarce full-documented cases of application.
- Despite CRISP-DM being found as the most used and adapted SDLC for BDAS, its application in the context of small-medium business projects can be highly cumbersome in organizational and technical demanded resources.
- The TDSP was found to have an overall HIGH conformance level, from this structural review of roles, phases-activities and work products, and assessed in the agile zone (+2.0 of score). However, this SDLC needs specific adequations to fit agile terminology of phases-activities and work products, as well as of more detailed activities guides and templates for work products. Additionally, despite there being a Sprint Plan, it is missing a Product Backlog Plan.
- The DDSL was confirmed as a lightweight SDLC but not as an agile one.

5. Conclusions

In this research, we applied a research method using a selective manual search of SDLCs for BDAS collected in the primary literature [30, 40, 42, 47, 49, 67, 68] on the basic literature of BDAS and Software Engineering, regarding the availability of agile SDLC for BDAS. This led us to select three SDLCs for BDAS: CRISP-DM, which is the most widely used SDLC today for developing BDAS projects [31]; TDSP, which is a claimed agile SDLC developed by Microsoft for BDAS projects based on CRISP-DM [51] that improves aspects of CRISP-DM; and a new lightweight SDLC, DDSL [24]. This review and evaluation were conducted by the research team, composed of a doctoral student, three full-time professors in the field of software engineering, and a full-time professor in the analytics data Science discipline.

Based on the results obtained in Tables 6 and 7, the following theoretical and practical contributions can be established:

- Theoretical contribution 1. Research on new SDLCs for BDAS has been practically null in the Software Engineering discipline in the 2000-2023 period in the literature consulted. The three SDLCs for BDAS were in the gray literature.
- Practical contribution 1. The three analyzed SDLCs (CRISP-DM, TDSP, and DDSL) are proprietary, and their public free-access documentation is limited.
- Practical contribution 2. TDSP was evaluated with a HIGH conformance level, but the analysis conducted on the content of the SDLC structure on roles, phases-activities, and artifacts, revealed that full-documented descriptions are missed. Thus, the application of TDSP as an agile Scrum-XP SDLC still needs adequation.

- Theoretical contribution 2. Although the main literature consulted [7, 8, 12, 22, 48, 74] on BDAS is adequately reported, and the topic is still relevant to business organizations today, we did not find an SDLC that can be considered a de facto standard as Rational Unified Process (RUP) was for software systems for two decades. CRISP-DM can be considered the most widely used and potentially converted into the de facto standard but for heavyweight BDAS projects.
- Practical contribution 3. For BDAS developers interested in using a lightweight SDLC, the recommendation is to use the DDSL SDLC.
- Practical contribution 4. For BDAS developers interested in agile approaches, the recommended SDLC is TDSP, but it requires adequations to fit the expected theoretical Scrum-XP SDLC.
- Based on the results obtained (Tables 6 and 7), the following recommendations for future research can also be made:
- Research on rigorous SDLC for BDAS is not encouraged, given the interest and need for agile and lightweight approaches at present.
- Conceptual research on agile SDLC for BDAS is encouraged to move towards an SDLC for BDAS that directly fits the theoretical Scrum-XP SDLC without the need to make additional adjustments, and that can be accepted and endorsed by the academic community.
- Both conceptual and empirical research on specific types of BDAS projects adequate for agile SDLC vs lightweight SDLC is required in the Software Engineering discipline, to establish core similarities and differences.
- To advance research on ISO/IEC standards for agile SDLC for BDAS in the context of small-medium business projects is required.
- Finally, we report the following methodological limitations of our study:
- This research focused only on the 5 development cycles for BDAS reported in at least 1 of the 3 comprehensive articles on big data development cycles. One development cycle was discarded for not meeting the characteristics of a methodology, and another for not being agile or lightweight.
- Only development cycles for agile-type BDAS projects were considered, and as a historical reference, the main methodology for developing this type of project is CRISP-DM (which is a heavyweight methodology). A lightweight methodology was added due to the scarcity of agile methodologies, and two other recent hybrid (agile/lightweight) methodologies, IBM ASUM and DDS, were discarded due to the minimal availability of academic references.
- This study used the agile framework Scrum-XP as the conceptual analysis framework, but future studies could consider another agile conceptual framework.
- This study analyzed the 3 methodologies exclusively using the original materials conceptually, without adding empirical evidence of their use by practitioners.
- The conceptual analysis was conducted by a research team composed of 1 final-year doctoral student; 3 senior professors in the field of software engineering (2 specialized in agile software engineering frameworks, and 1 specialized in software engineering processes); and 1 professor specialized in data science, with an average combined academic and research experience of 14 years. We believe that a research team with similar demographic characteristics would arrive at similar conclusions.

Hence, we can indicate that there is a need to achieve better agile SDLCs for BDAS that can be supported theoretically and used in practice (i.e., with high levels of usability, ease of use, compatibility, and perceived value by BDAS developers) for the small-medium organizations. Therefore, further conceptual, and empirical research is encouraged in these relevant research streams.

Table 7. Evaluation of CRISP-DM, TDSP, DDSL, and Scrum-XP SDLC for BDAS using the rigorous-agile SDLC framework of 7 attributes.

Rigor attributes	LEVEL ASSIGNED TO THE SDLC FOR BDAS							Agility attributes
	Zones of rigorous SDLCs		Zones of lightweight SDLCs			Zones of agile SDLCs		
	-3 Very high	-2 High	-1 Low	0 Neutral	+1 Low	+2 High	+3 Very high	
Rigid: to keep and apply BDAS practices without any variation.		CRISP-DM		DDSL	TDSP		Scrum-XP	Flexible: to reconfigure BDAS practices when necessary.
Bureaucratic: to ignore unexpected events during the BDAS development process accepting potential negative consequences.		CRISP-DM		DDSL	TDSP		Scrum-XP	Responsive: to sense the environment and react appropriately to unexpected events during the BDAS development process.
Slow: to deliver a usable BDAS in relatively large periods.		CRISP-DM		DDSL	TDSP		Scrum-XP	Speedy: to deliver quickly a usable BDAS.
Sophisticated: to pursue the best designed and built BDAS.		CRISP-DM		DDSL	TDSP		Scrum-XP	Lean: to pursue a minimum viable BDAS (that could be incremented in the next releases).
Hard: high cognitive load and high training effort to be learned and used		CRISP-DM		DDSL	TDSP		Scrum-XP	Simple: low cognitive load and low training effort to be learned and used.
Heavyweight: high volume of practices.		CRISP-DM		DDSL	TDSP		Scrum-XP	Lightweight: shortened practices from the original heavyweight practices but still considered useful for agile domains.
Mandatory documentation: it demands the fulfillment of mandatory technical and user documentation.		CRISP-DM		DDSL	TDSP		Scrum-XP	Optional documentation: it permits the fulfillment of technical and user documentation.
Overall level		CRISP-DM		DDSL	TDSP		Scrum-XP	

Table 8. Analysis of CRISP-DM, TDSP, and DDSL against the theoretical Scrum-XP SDLC for BDAS.

SDLC element	Theoretical generic Scrum-XP SDLC Pro Forma	CRISP-DM	TDSP	DDSL
Roles (3)	User roles: 1. Management roles: 1. Technical roles: 1.	User roles: 1. Customer. Management roles: 1. Project Manager. Technical roles: 1. Developer Team.	User roles: • Role 0. Customer. Management roles: • Role 1. Group manager. • Role 2. Team lead. • Role 3. Project lead. Technical roles: • Role 4. Project individual contributors (data scientists, business analysts, data engineers, solution architect, application developers)	User roles: • Role 1. Business stakeholder Management roles: • Role 2. Data product manager Technical roles: • Role 3. Data scientist • Role 4. Data infrastructure engineer. • Role 5. Data storyteller.
Phases-activities (6, 13)	Pre-game phases: • Phase 1. Product exploration: (with 3 activities). • Phase 2. Product release planning: (with 1 activity).	• Phase 1. Business understanding. Activities 1: 1. Determine business objectives. 2. Assess situation. 3. Determine data mining goals. 4. Produce project plan. • Phase 2. Data understanding. Activities 2: 1. Collect initial data. 2. Describe data. 3. Explore data. 4. Verify data quality. • Phase 3. Data preparation. Activities 3: 1. Select data. 2. Clean data. 3. Construct data. 4. Integrate data. 5. Format data.	• Phase 1. Business understanding: Activities 1: 1. Define Objectives. 2. Identify data sources. • Phase 2. Data acquisition and understanding: Activities 2: 1. Ingest the Data. 2. Explore the Data. 3. Set up a Data Pipeline.	• Phase.1 Ideation: Activities 1: 1. Identified problem. 2. Project Scoping. a) Review prior art. b) Calculate value. c) Assess feasibility. 3. Manage backlog. 4. Select artifacts. • Phase 2. Data acquisition and exploration. Activities 2: 1. Getting the data. 2. Identify Sources the data. a) Connect. 3. Create data (capture). 4. Buy and ingest data. 5. Explore data. 6. Prepare data.
	Game Phases: • Phase 3. Sprint-iteration planning: (with 1 activity). • Phase 4. Sprint-iteration development: (with 5 activities). • Phase 5. Sprint-iteration review and retrospective: (with 2 activities).	• Phase 4. a) Conceptual modeling. Activities 4-a): 1. Select modeling techniques. 2. Generate test design. • Phase 4. b) Computational modeling. Activities 4-b): 3. Build model. 4. Assess model. • Phase 6. Evaluation. Activities 5: 1. Evaluate results. 2. Review process. 3. Determine next Steps.	• Phase 0. Agile project management: Activities 0: 1. Plan Sprint. 2. Review code built from several branches. 3. Merge-delete branches. • Phase 3. Modeling: Activities 3: 1. Feature engineering. 2. Model training. 3. Model evaluation. • Phase 4. Deployment: Activities 4: 1. Operationalize a model.	• Phase 3. Research and development: Activities 3: 1. Generate Hypothesis. 2. Validate right tools. a) IT request. b) Experiment. c) Assess result. 3. Validate the need new data, insightful? 4. Share insight. • Phase 4. Validation. Activities 4: 1. Validate the business. 2. Validate technically. 3. Validate ready to deploy. 4. Publish.
	Post-game phase: • Phase 6. Product release: (with 1 activity).	• Phase 6. Deployment. Activities 6: 5. Plan Deployment. 6. Plan monitoring and maintenance. 7. Produce final report. 8. Review project.	• Phase 5. Customer acceptance. Activities 5: 1. System Validation. 2. Project hand-off.	• Phase 5. Delivery. Activities 5: 1 Plan Delivery 2 Deploy 3 Test • Phase 6. Monitoring: Activities 6: 1. Monitor. a) Usage. b) Performance. c) Value. 2. Identify improvements. 3. Generate value.
Work products (15)	Pre-game phases: • Phase 1. Product exploration: (3 work products). • Phase 2. Product release planning: (1 work products).	• Phase 1. Business understanding. Work products 1: 1. Background. 2. Business objectives. 3. Business success criteria. 4. Inventory of resources.	• Phase 1. Business understanding. Work products 1: 2. Data source. 3. Data Dictionaries. • Phase 2. Data acquisition and understanding.	• Phase 1. Ideation. Work products 0: 1. Project Scope document. 1. Project Scope document. 2. Project Kick-off. Work products 1:

		5. Requirements, assumptions, and constraints. 6. Risks and contingencies. 7. Terminology. 8. Costs and benefits. 9. Data mining goals. 10. Data mining success criteria 11. Project plan. 12. Initial assessment of tools and techniques. • Phase 2. Data understanding. Work products 2: 1. Initial data collection report. 2. Data Description report. 3. Data exploration report. 4. Data quality report. • Phase 3. Data preparation. Work products 3: 1. Rationale for inclusion/exclusion. 2. Data cleaning report. 3. Derived attributes. 4. Generated records. 5. Merged data. 6. Reformatted data. 7. Dataset. 8. Dataset description.	1. Data quality report. 2. Solution architecture. 3. Checkpoint decision.	1. Model requirements Doc. • Phase 2. Data acquisition and exploration. Work products 2: 2. Data dictionary.
	Game Phases: • Phase 3. Sprint-iteration planning: (1 work product). • Phase 4. Sprint-iteration development: (8 work products). • Phase 5. Sprint-iteration review and retrospective: (1 work product).	• Phase 4. a) Conceptual modeling. 1. Modeling technique. 2. Modeling Assumptions. 3. Test design. • Phase 4. b) Computational modeling: 4. Parameter settings. 5. Models. 6. Model descriptions. 7. Model assessment. 8. Revised parameter Settings. • Phase 5. Evaluation: 1. Assessment of data mining results. 2. Approved models. 3. Review of process. 4. List of possible actions. 5. Decision.	• Phase 0. TDSP workflow for project execution. Work products 0: 1. Sprint Plan. Work products 1: 1. Charter Document. • Phase 3. Modeling: Work products 3: 1. Model. • Phase 4. Deployment. Work products 3: 1. A status Dashboard that displays the system health and key metrics. 2. A final modeling report with deployment details. 3. A final solution architecture document.	• Phase 3. Research and development. Work products 3: *Data model experiment • Phase 4. Validation. Work products 4: *Validated data model.
	Post-Game Phase: • Phase 6. Product release: (1 work product).	• Phase 6. Deployment. 1. Deployment Plan. 2. Monitoring and Maintenance Plan. 3. Final Report. 4. Final Presentation. 5. Experience Documentation.	• Phase 5. Customer acceptance. Work products 5: 1. Exit report of the project for the customer.	• Phase 5. Delivery. Work products 5: *Production Data Model. • Phase 6. Monitoring. Work products 6: Monitoring and training plan.

References

[1] Abrahamsson P., Oza N., and Siponen M., *Agile Software Development: Current Research and Future Directions*, Springer, 2010. https://link.springer.com/chapter/10.1007/978-3-642-12575-1_3

[2] Ahimbisibwe A., Daellenbach U., and Cavana R., "Empirical Comparison of Traditional Plan-Based and Agile Methodologies: Critical Success Factors for Outsourced Software Development Projects from Vendors' Perspective," *Journal of Enterprise Information Management*, vol. 30, no. 3, pp. 400-453, 2017. https://doi.org/10.1108/JEIM-06-2015-0056

[3] Ajah I. and Nweke H., "Big Data and Business Analytics: Trends, Platforms, Success Factors and Applications," *Big Data and Cognitive Computing*, vol. 3, no. 2, pp. 1-30, 2019. https://www.mdpi.com/2504-2289/3/2/32

[4] Alsaqqa S., Sawalha S., and Abdel-Nabi H., "Agile Software Development: Methodologies and Trends," *International Journal of Interactive Mobile Technologies*, vol. 14, no. 11, pp. 246-270, 2020. https://doi.org/10.3991/ijim.v14i11.13269

[5] Andoh-Baidoo F., Baker E., Susarapu S., and Kasper G., "A Review of IS Research Activities and Outputs Using Pro Forma Abstracts," *Information Resources Management Journal*, vol. 20, no. 4, pp. 65-79, 2007. https://www.igi-

global.com/article/review-research-activities-outputs-using/1

[6] Andoh-Baidoo F., Chavarria J., Jones M., Wang Y., and Takieddine S., "Examining the State of Empirical Business Intelligence and Analytics Research: A Poly-Theoretic Approach," *Information and Management*, vol. 59, no. 6, pp. 103677, 2022. https://doi.org/10.1016/j.im.2022.103677

[7] Batra D., Xia W., VanderMeer D., and Dutta K., "Balancing Agile and Structured Development Approaches to Successfully Manage Large-Distributed Software Projects: A Case Study from the Cruise Line Industry," *Communications of the Association for Information Systems*, vol. 27, pp 379-395, 2010. https://aisel.aisnet.org/cais/vol27/iss1/21/

[8] Beck K., "Embracing Change with extreme Programming," *Computer*, vol. 32, no. 10, pp. 70-77, 1999. https://dl.acm.org/doi/10.1109/2.796139

[9] Beck K., Beedle M., Van Bennekum A., Cockburn A., Cunningham W., Fowler M., Grenning J., Highsmith J., Hunt A., Jeffries R., Kern J., Marick B., Martin R., Mellor S., Schwaber K., Sutherland J., and Thomas D., The Agile Manifesto, 2001, https://agilemanifesto.org/, Last Visited, 2042.

[10] Beulke D., Big Data Impacts Data Management: The 5 Vs of Big Data, 2011, https://davebeulke.com/big-data-impacts-data-management-the-five-vs-of-big-data/, Last Visited, 2024.

[11] Boehm B. and Turner R., "Management Challenges to Implementing Agile Processes in Traditional Development Organizations," *IEEE Software*, vol. 22, no. 5, pp. 30-39, 2005. https://ieeexplore.ieee.org/document/1504661

[12] Boehm B. and Turner R., "Using Risk to Balance Agile and Plan-Driven Methods," *Computer*, vol. 36, no. 6, pp. 57-66, 2003. https://ieeexplore.ieee.org/document/1204376

[13] Boudali I., Chebaane S., and Zitouni Y., "A Predictive Approach for Myocardial Infarction Risk Assessment Using Machine Learning and Big Clinical Data," *Healthcare Analytics*, vol. 5, pp. 100319, 2024. https://doi.org/10.1016/j.health.2024.100319

[14] Bourque P. and Fairly R., *SWEBOK Version 3.0-Guide to the Software Engineering Body of Knowledge*, IEEE, 2014. https://ieeecs-media.computer.org/media/education/swebok/swebok-v3.pdf

[15] Campanelli A. and Parreiras F., "Agile Methods Tailoring-A Systematic Literature Review," *Journal of Systems and Software*, vol. 110, pp. 85-100, 2015. https://doi.org/10.1016/j.jss.2015.08.035

[16] Chapman P., Clinton J., Kerber R., Khabaza T., Reinartz T., Shearer C., and Wirth R., *CRISP-DM 1.0-Step-by-Step Data Mining Guide*, SPSS Inc., 2000. https://mineracaodedados.wordpress.com/wp-content/uploads/2012/12/crisp-dm-1-0.pdf

[17] Conboy K., "Agility from First Principles: Reconstructing the Concept of Agility in Information Systems Development," *Information Systems Research*, vol. 20, no. 3, pp. 329-354, 2009. https://pubsonline.informs.org/doi/10.1287/isre.1090.0236

[18] Cox M. and Ellsworth D., "Managing Big Data for Scientific Visualization," *ACM Sig-Graph, MRJ/NASA Ames Res, Center*, vol. 97, no. 1, pp. 21-38, 1997. https://www.researchgate.net/publication/238704525_Managing_big_data_for_scientific_visualization

[19] Data Science for all, Analytics Solutions Unified Method for Data Mining, IBM, 2015, https://datascienceforall.wordpress.com/data-mining-and-predictive-analytics/, Last Visited, 2024.

[20] Davenport T. and Bean R., Data and AI Leadership Executive Survey, Data and A Leader Ship Executive Survey 2022, https://wwa.wavestone.com/en/insight/data-ai-leadership-executive-survey-2022/, Last Visited, 2024.

[21] Davenport T. and Malone K., "Deployment as a Critical Business Data Science Discipline," *Harvard Data Science Review*, vol. 3, no. 1, pp. 1-11, 2021. https://doi.org/10.1162/99608f92.90814c32

[22] Digital.AI, 16th Annual State of Agile Report, 2022, https://digital.ai/resource-center/analyst-reports/16th-state-of-agile-report/, Last Visited, 2024.

[23] Dingsoyr T., Neru S., Balijepally V., and Moe N., "A Decade of Agile Methodologies: Towards Explaining Agile Software Development," *Journal of Systems and Software*, vol. 85, no. 6, pp. 1213-1221, 2012. https://doi.org/10.1016/j.jss.2012.02.033

[24] Domino Data Lab, The Practical Guide to Managing Data Science at Scale (2017), https://domino.ai/resources/managingdatascienc, Last Visited, 2024.

[25] Dudziak T., "Extreme Programming an Overview," *Methoden und Werkzeuge der Softwareproduktion WS*, vol. 1, no. 28, pp. 1-28, 2000. https://csis.pace.edu/~marchese/CS616/Agile/XP/XP_Overview.pdf

[26] Dyba T. and Dingsøyr T., "Empirical studies of Agile Software Development: A Systematic

Review," *Information and Software Technology*, vol. 50, no. 9-10, pp. 833-859, 2008. https://doi.org/10.1016/j.infsof.2008.01.006

[27] Fayyad U., Haussler D., and Stolorz P., "KDD for Science Data Analysis: Issues and Examples," *in Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, Portland, pp. 50-56, 1996. https://dl.acm.org/doi/abs/10.5555/3001460.3001471

[28] Giray G., "A Software Engineering Perspective on Engineering Machine Learning Systems: State of the Art and Challenges," *Journal of Systems and Software*, vol. 180, pp. 111031, 2021. https://doi.org/10.1016/j.jss.2021.111031

[29] Grey E., Jennings W., Farrall S., and Hay C., "Small Big Data: Using Multiple Data-Sets to Explore Unfolding Social and Economic Change," *Big Data and Society*, vol. 2, no. 1, 2015. https://doi.org/10.1177/2053951715589418

[30] Haakman M., Cruz L., Huijgens H., and Van Deursen A., "AI Lifecycle Models Need to be Revised: An Exploratory Study in Fintech," *Empirical Software Engineering*, vol. 26, no. 5, pp. 1-29, 2021. https://doi.org/10.1007/s10664-021-09993-1

[31] Halper F., "Next-Generation Analytics and Platforms for Business Success," *TDWI, Research Report*, 2015. https://tdwi.org/webcasts/2015/01/nextgeneration-analyticsand-platforms-for-business-success.aspx

[32] Highsmith J. and Cockburn A., "Agile Software Development: The Business of Innovation," *Computer*, vol. 34, no. 9, pp. 120-27, 2001. https://ieeexplore.ieee.org/document/947100

[33] Hobbs B. and Petit Y., "Agile Methods on Large Projects in Large Organizations," *Project Management Journal*, vol. 48, no. 3, 3-19, 2017. https://doi.org/10.1177/875697281704800301

[34] Hoda R., Salleh N., and Grundy J., "The Rise and Evolution of Agile Software Development," *IEEE Software*, vol. 35, no. 5, pp. 58-63, 2018. https://ieeexplore.ieee.org/document/8409911

[35] Iranmanesh M., Lim K., Foroughi B., Hong M., and Ghobakhloo M., "Determinants of Intention to Adopt Big Data and Outsourcing among SMEs: Organizational and Technological Factors as Moderators," *Management Decision*, vol. 61, no. 1, pp. 201-222, 2023. https://doi.org/10.1108/MD-08-2021-1059

[36] Jones M., Big Data is a 'New Natural Resource' IBM Says, 2012, http://www.govtech.com/policymanagement/BigDataIsaNewNaturalResourceIBMSays.html, Last Visited, 2024.

[37] Kitchenham B., Brereton O., Budgen D., Turner M., Bailey J., and Linkman S., "Systematic Literature Reviews in Software Engineering-A Systematic Literature Review," *Information and Software Technology*, vol. 51, no. 1, pp. 7-15, 2009. https://doi.org/10.1016/j.infsof.2008.09.009

[38] Kitchin R. and Lauriault T., "Small Data in the Era of Big Data," *Geo Journal*, vol. 80, no. 4, pp. 463-475, 2015. https://www.jstor.org/stable/44076310

[39] Klotins E., Unterkalmsteiner M., Chatzipetrou P., Gorschek T., Prikladnicki R., Tripathi N., and Pompermaier L., "Use of Agile Practices in Start-Up Companies," *e-Informatica Software Engineering Journal*, vol. 15, no. 1, 2021. DOI:10.37190/e-Inf210103

[40] Kumar V. and Alencar P., "Software Engineering for Big Data Projects: Domains, Methodologies and Gaps," *in Proceedings of the IEEE International Conference on Big Data*, Washington (DC), pp. 2886-2895, 2016. DOI:10.1109/BigData.2016.7840938

[41] Kune R., Konugurthi P., Agarwal A., Chillarige R., and Buyya R., "The Anatomy of Big Data Computing," *Software: Practice and Experience*, vol. 46, no. 1, pp. 79-105, 2016. https://onlinelibrary.wiley.com/doi/10.1002/spe.2374

[42] Laigner R., Kalinowski M., Lifschitz S., Monteiro R., and De Oliveira D., "A Systematic Mapping of Software Engineering Approaches to Develop Big Data Systems," *in Proceedings of the 44th Euromicro Conference on Software Engineering and Advanced Applications*, Prague, pp. 446-453, 2018. DOI:10.1109/SEAA.2018.00079

[43] Laporte C. and O'Connor R., "Systems and Software Engineering Standards for very Small Entities: Accomplishments and Overview," *Computer*, vol. 49, no. 8, pp. 84-87, 2016. https://ieeexplore.ieee.org/document/7543423

[44] Larson D. and Chang V., "A Review and Future Direction of Agile, Business Intelligence, Analytics and Data Science," *International Journal of Information Management*, vol. 36, no. 5, pp. 700-710, 2016. https://doi.org/10.1016/j.ijinfomgt.2016.04.013

[45] Lin Y. and Huang S., "The Design of a Software Engineering Lifecycle Process for Big Data," *IT Professional*, vol. 20, no. 1, pp. 45-52, 2018. DOI:10.1109/MITP.2018.011291352

[46] Lukoianova T. and Rubin V, "Veracity Roadmap: Is Big Data Objective, Truthful and Credible?," *Advances in Classification Research Online*, vol. 24, no. 1, pp. 4-15, 2014. https://journals.lib.washington.edu/index.php/acro/article/view/14671

[47] Madhavji N., Miranskyy A., and Kontogiannis K., "Big Picture of Big Data Software Engineering: with Example Research Challenges," *in Proceedings of the IEEE/ACM 1st International*

Workshop on Big Data Software Engineering, Florence, pp. 11-14, 2015. DOI: 10.1109/BIGDSE.2015.10

[48] Magdaleno A., Werner C., and De Araujo R., "Reconciling Software Development Models: A Quasi-Systematic Review," *Journal of Systems and Software*, vol. 85, no. 2, pp. 351-369, 2012. https://doi.org/10.1016/j.jss.2011.08.028

[49] Martinez I., Viles E., and Olaizola I., "Data Science Methodologies: Current Challenges and Future Approaches," *Big Data Research*, vol. 24, pp. 100183, 2021. https://doi.org/10.1016/j.bdr.2020.100183

[50] Martinez-Plumed F., Contreras-Ochando L., Ferri C., Hernandez-Orallo J., Kull M., Lachiche N., Ramirez-Quintana M., and Flach P., "CRISP-DM Twenty Years Later: From Data Mining Processes to Data Science Trajectories," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 8, pp. 3048-3061, 2021. DOI:10.1109/TKDE.2019.2962680

[51] Microsoft Learn, What is the Team Data Science Process?, https://docs.microsoft.com/enus/azure/machinelearning/teamdatascienceprocess/overview, Last Visited, 2024.

[52] Montoya-Murillo D., Mora M., Galvan-Cruz S., and Munoz-Zavala A., *Development Methodologies for Big Data Analytics Systems: Plan-driven, Agile, Hybrid, Lightweight Approaches*, Springer, 2023. https://link.springer.com/chapter/10.1007/978-3-031-40956-1_5

[53] Mora M., Adelakun O., Galvan-Cruz S., and Wang F., "Impacts of IDEF0-Based Models on the Usefulness, Learning, and Value Metrics of Scrum and XP Project Management Guides," *Engineering Management Journal*, vol. 34, no. 4, pp. 574-590, 2021. https://doi.org/10.1080/10429247.2021.1958631

[54] Mora M., Adelakun O., Reyes-Delgado P., and Diaz O., "AVS_FD_MVITS: An Agile IT Service Design Workflow for Small Data Centers," *The Journal of Supercomputing*, vol. pp. 17519-17561, 2023. https://link.springer.com/article/10.1007/s11227-023-05244-w

[55] Mora M., Reyes-Delgado P., Galvan-Cruz S., and Solano-Romo L., *Development Methodologies for Big Data Analytics Systems: Plan-driven, Agile, Hybrid, Lightweight Approaches*, Springer, 2024, https://link.springer.com/chapter/10.1007/978-3-031-40956-1_1

[56] Mora M., Wang F., Gomez J., and Diaz O., *Trends and Applications in Software Engineering*, Springer, 2020. https://link.springer.com/chapter/10.1007/978-3-030-33547-2_9

[57] Oussous A., Benjelloun F., Lahcen A., and Belfkih S., "Big Data Technologies: A Survey," *Journal of King Saud University-Computer and Information Sciences*, vol. 30, no. 4, pp. 431-448, 2018. https://doi.org/10.1016/j.jksuci.2017.06.001

[58] Paakkonen P. and Pakkala D., "Reference Architecture and Classification of Technologies, Products and Services for Big Data Systems," *Big Data Research*, vol. 2, no. 4, 166-186, 2015. https://doi.org/10.1016/j.bdr.2015.01.001

[59] Phillips-Wren G., Daly M., and Burstein F., "Reconciling Business Intelligence, Analytics and Decision Support Systems: More Data, Deeper Insight," *Decision Support Systems*, vol. 146, pp. 113560, 2021. https://doi.org/10.1016/j.dss.2021.113560

[60] Pino F., Pedreira O., García F., Luaces M., and Piattini M., "Using Scrum to Guide the Execution of Software Process Improvement in Small Organizations," *Journal of Systems and Software*, vol. 83, no. 10, 1662-1677, 2010. https://doi.org/10.1016/j.jss.2010.03.077

[61] Pollack J., Helm J., and Adler D., "What is the Iron Triangle, and how has it Changed?," *International Journal of Managing Projects in Business*, vol. 11, no. 2, pp. 527-547, 2018. https://www.emerald.com/insight/content/doi/10.1108/IJMPB-09-2017-0107/full/html

[62] Qumer A. and Henderson-Sellers B., "An Evaluation of the Degree of Agility in Six Agile Methods and its Applicability for Method Engineering," *Information and Software Technology*, vol. 50, no. 4, pp. 280-295, 2008. https://doi.org/10.1016/j.infsof.2007.02.002

[63] Ransbotham S., Khodabandeh S., Kiron D., Candelon F., Chu M., and LaFountain B., "Expanding AI's Impact with Organizational Learning," *MIT Sloan Management Review and Boston Consulting Group*, pp. 1-15, 2020. https://sinnergiak.org/2021/01/18/ampliando-el-impacto-de-la-ia-con-el-aprendizaje-organizacional/?lang=en

[64] Rao T., Mitra P., Bhatt R., and Goswami A., "The Big Data System, Components, Tools, and Technologies: A Survey," *Knowledge and Information Systems*, vol. 60, no. 3, pp. 1165-1245, 2019. https://link.springer.com/article/10.1007/s10115-018-1248-0

[65] Russom P., Big Data Analytics, 2011, https://origin-tableau-www.tableau.com/sites/default/files/whitepapers/tdwi_bpreport_q411_big_data_analytics_tableau.pdf, Last Visited, 2024.

[66] Salazar-Salazar G., Mora M., Duran-Limon H., and Rodriguez F., *Development Methodologies for Big Data Analytics Systems: Plan-driven,*

Agile, Hybrid, Lightweight Approaches, Springer, 2023. https://link.springer.com/chapter/10.1007/978-3-031-40956-1_6

[67] Saltz J. and Krasteva I., "Current Approaches for Executing Big Data Science Projects-A Systematic Literature Review," *PeerJ Computer Science*, vol. 8, pp. 862, 2022. https://doi.org/10.7717/peerj-cs.862

[68] Saltz J. and Shamshurin I., "Big Data Team Process Methodologies: A Literature Review and the Identification of Key Factors for a Project's Success," *in Proceedings of the IEEE International Conference on Big Data*, Washington (DC), pp. 2872-2879, 2016. DOI:10.1109/BigData.2016.7840936

[69] Saltz J., Data Driven Scrum, 2022, https://www.datascience-pm.com/data-driven-scrum/, Last Visit, 2024.

[70] Schryen G., "Writing Qualitative IS Literature Reviews-Guidelines for Synthesis, Interpretation, and Guidance of Research," *Communications of the Association for Information Systems*, vol. 37, no. 1, pp. 286-325, 2015. https://aisel.aisnet.org/cais/vol37/iss1/12/

[71] Schwaber K. and Mar K., Scrum with XP (2002), https://www.informit.com, Last Visited, 2024.

[72] Schwaber K. and Sutherland J., The Scrum Guide (2020), https://scrumguides.org/, Last Visited, 2024.

[73] Schwaber K., "Scrum Development Process," *in Proceedings of the Business Object Design and Implementation*, Austin, pp. 117-134, 1997. https://link.springer.com/chapter/10.1007/978-1-4471-0947-1_11

[74] Sutherland J., *The Scrum Handbook*, Scrum Training Institute Press, 2010. https://www.researchgate.net/publication/301685699_Jeff_Sutherland's_Scrum_Handbook

[75] Taranum A., Metan J., Yogegowda P., and Krishnappa C., "Canine Disease Prediction using Multi-Directional Intensity Proportional Pattern with Correlated Textural Neural Network," *The International Arab Journal of Information Technology*, vol. 21, no. 5, pp. 899-914, 2024. doi:10.34028/iajit/21/5/11

[76] Tell P., Klunder J., Kupper S., Raffo D., MacDonell S., Munch J., Pfahl D., Linssen O., and Kuhrmann M., "Towards the Statistical Construction of Hybrid Development Methods," *Journal of Software: Evolution and Process*, vol. 33, no. 1, pp. 2315, 2021. https://doi.org/10.1002/smr.2315

[77] Todman L., Bush A., and Hood A., "Small Data' for Big Insights in Ecology," *Trends in Ecology and Evolution*, vol. 38, no. 7, pp. 615-622, 2023. https://www.cell.com/trends/ecology-evolution/fulltext/S0169-5347(23)00019-8

[78] Tsai C., Lai C., Chao H., and Vasilakos A., "Big Data Analytics: A Survey," *Journal of Big Data*, vol. 2, no. 1, pp. 1-32, 2015. https://doi.org/10.1186/s40537-015-0030-3

[79] Tsoy M. Staples D., "What are the Critical Success Factors for Agile Analytics Projects?," *Information Systems Management*, vol. 38, no. 4, pp. 324-341, 2021. https://doi.org/10.1080/10580530.2020.1818899

[80] Vallon R., Da Silva Estacio B., Prikladnicki R., and Grechenig T., "Systematic Literature Review on Agile Practices in Global Software Development," *Information and Software Technology*, vol. 96, pp. 161-180, 2018. https://doi.org/10.1016/j.infsof.2017.12.004

[81] Walker J., Big Data Strategies Disappoint with 85 Percent Failure Rate, Digital Journal, 2017, https://www.digitaljournal.com/tech- science/big-data-strategies-disappoint-with-85-percent-failure- rate/article/508325, Last Visited, 2024.

[82] Watson H., "Update Tutorial: Big Data Analytics: Concepts, Technology, and Applications," *Communications of the Association for Information Systems*, vol. 44, pp. 364-379, 2019. https://aisel.aisnet.org/cais/vol44/iss1/21/

[83] Wohlin C., Runeson P., Host M., Ohlsson M., Regnell B., and Wesslen A., *Experimentation in Software Engineering*, Springer, 2012. https://dl.acm.org/doi/book/10.5555/2349018

[84] Zdrenka W., "The Use and the Future of Big Data Analytics in Supply Chain Management," *Research in Logistics and Production*, vol. 7, no. 2, pp. 91-102, 2017. https://sin.put.poznan.pl/publications/details/i32187

Gerardo Salazar-Salazar is a Doctoral student in Applied and Technological Sciences, with a focus on the use of agile methodologies for Data Science projects to increase the success of Big Data Software Systems Projects in small companies. Professor with 1 year of experience in the Department of Electronic Systems of the Autonomous University of Aguascalientes. With a chapter published in the book Development Methodologies for Big Data Analytics Systems: Plan-Based, Agile, Hybrid and Lightweight Approaches.

Manuel Mora received the M.Sc. Degree in Artificial Intelligence from Monterrey Tech, in 1989, and the Eng.D. Degree in engineering from the National Autonomous University of Mexico (UNAM), in 2003. He has been a full-time Professor with the Autonomous University of Aguascalientes (UAA), Aguascalientes, Mexico, since 1994. He has published over 100 research papers in international top conferences, research books, refereed journals listed in JCRs and Scopus indexes, and co-edited five international research books on the topics of DMSS, IT services and data centers, and research methods. His research articles have been published in JCR indexed journals, such as IEEE Transactions on Systems, Man, and Cybernetics: Systems, European Journal of Operational Research, International Journal of Information Management, Engineering Management Journal, International Journal of Information Technology and Decision Making, Information Technology for Development, International Journal of Software Engineering and Knowledge Engineering, Computer Standards and Interface, Software Quality Journal, Expert Systems, Software and Systems Modeling, Journal of Organizational Computing and Electronic Commerce, and Journal of Supercomputing. His current research interests include agile development methodologies for: IT services, big data software systems, ontology-based KMS, and SOA/MSA-based software systems. He is also an ACM Senior Member and a Mexican National Researcher at Level II.

Hector Duran-Limon is currently a full Professor at the Information Systems Department at the University of Guadalajara. He completed a Ph.D. at Lancaster University in 2002. Following this, he was a post-doctoral researcher until December 2003. He obtained an IBM Faculty Award in 2008. His research interests include Grid Computing, Adaptive Middleware, and Mobile Computing. He is also interested in software architectures and component-based development. In 2006, He was invited to create a PhD program in Information Technologies for the University of Guadalajara, becoming a member of the Academic Council. Contact him at the Information Systems Department, University of Guadalajara, Mexico.

Francisco Alvarez-Rodriguez professor of Software Engineering, Department of Computer Science, University of Aguascalientes (U.A.A.). Ph.D. in Teaching Methodology from IMEP (Mexico). Ph.D. in Engineering from UNAM (Mexico). He has been Dean of the Center of Basic Sciences at the U.A.A., as well as Head of the Department of Electronic Systems. Member of academic cores of several graduate programs at the U.A.A. Doctorate in Computer Science, Interinstitutional Doctorate in Science, Master of Science with option in Mathematics and Computer Science. Author of books and articles on the line of Learning Objects and Software Development Processes. Member of the National System of Researchers (SNI) and International Research Groups. He is currently president of the National Accreditation Council of Computer Science and Informatics Programs, A.C.

Angel Munoz-Zavala obtained a Master's Degree and Doctorate in Computer Science, both from the Center for Research in Mathematics (CIMAT), Guanajuato, Mexico. Since August 2009, he is a professor-researcher at the Autonomous University of Aguascalientes. He is primarily interested in AI (Evolutionary Algorithms) and Statistical Science (Reliability Systems). He has published more than 20 articles in international journals and conferences on these topics. In 2009, he obtained 1st place in the Ph.D. thesis category by the Mexican AI Society.